

# E62

## Mastering the IMS OTMA Protocols and Messages

Jack Yuan

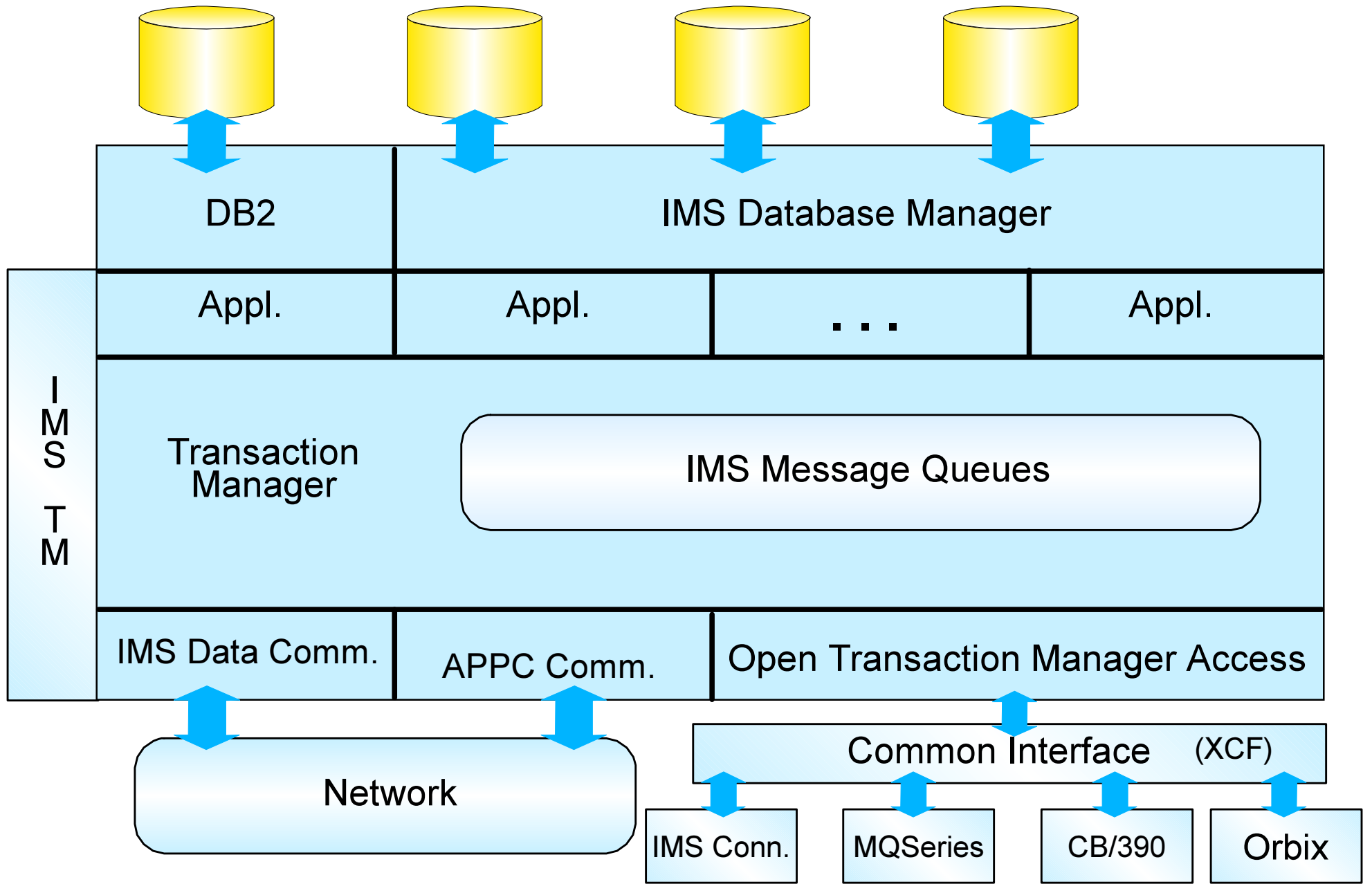


Anaheim, California

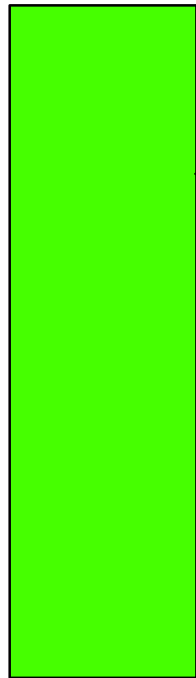
October 23 - 27, 2000

# Agenda

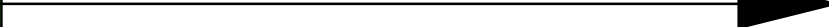
- IMS OTMA Intro
- OTMA message
- Client-bid protocol
- Basic protocol for message processing
- Send-then-commit protocol
- Commit-then-send protocol
- Program switches
- Resynch protocol
- OTMA Message Prefix
- IMS 01 and 6701 log records for OTMA



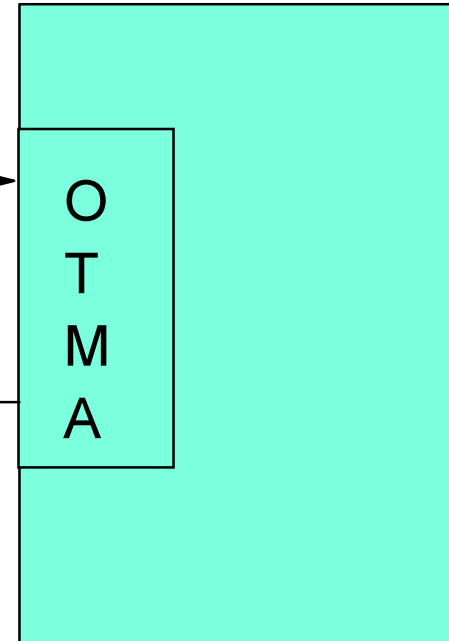
**OTMA  
Client**



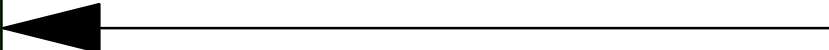
OTMA prefix +  
(application data)



**IMS**

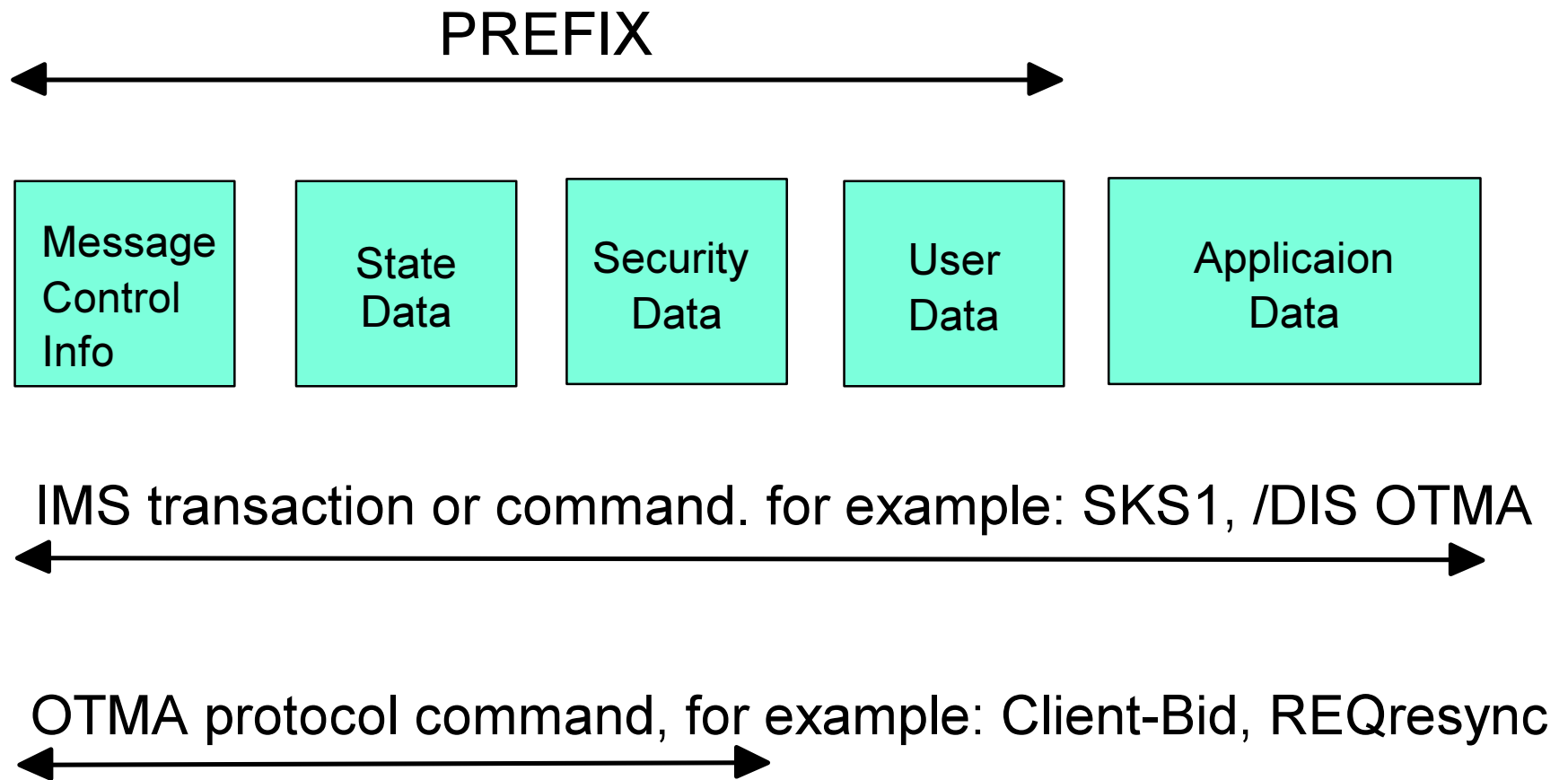


OTMA prefix +  
(application data)

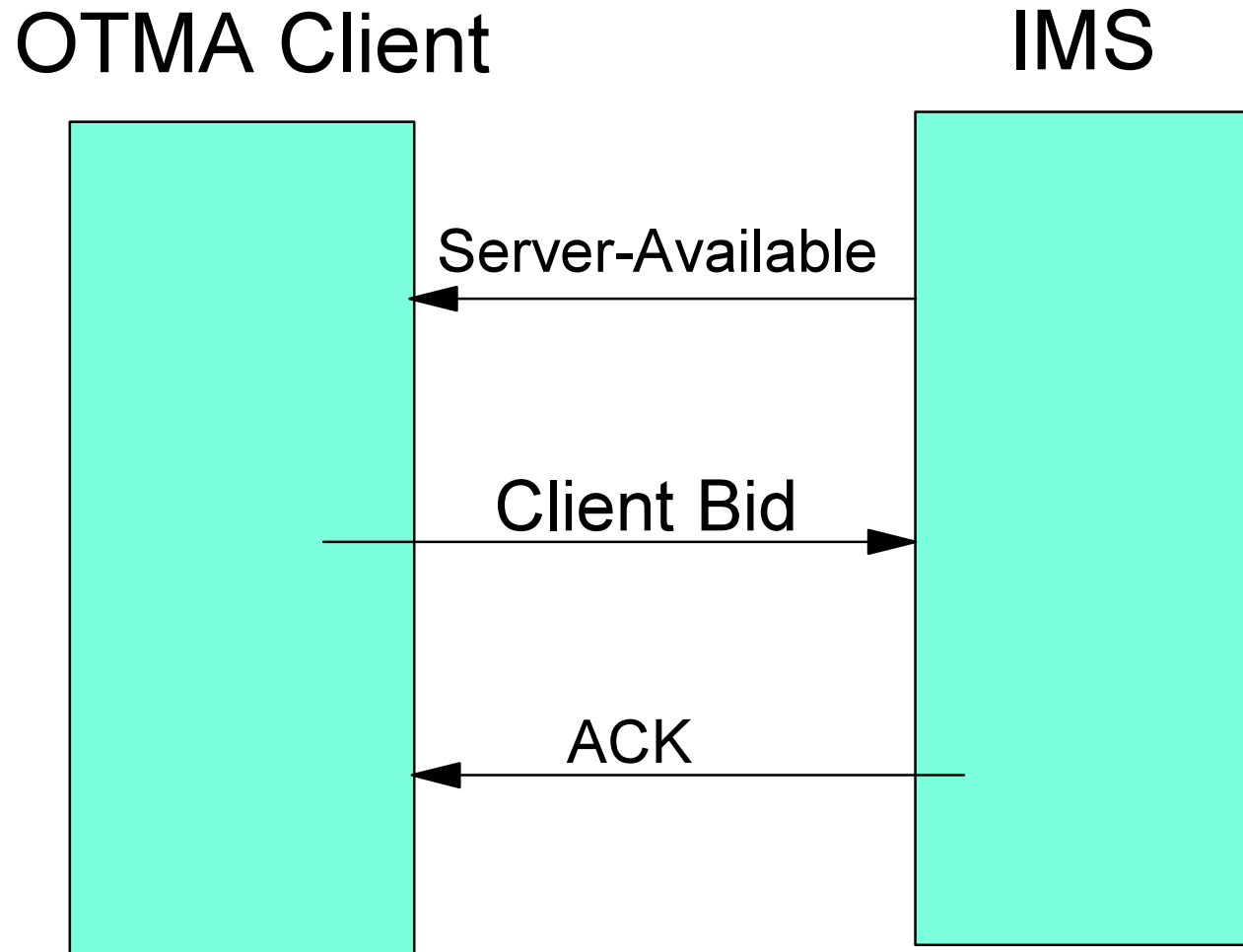


An IMS OTMA protocol command is an OTMA message without application data.

# OTMA Messages ( DFSYMSG )



# Client Bid Protocol

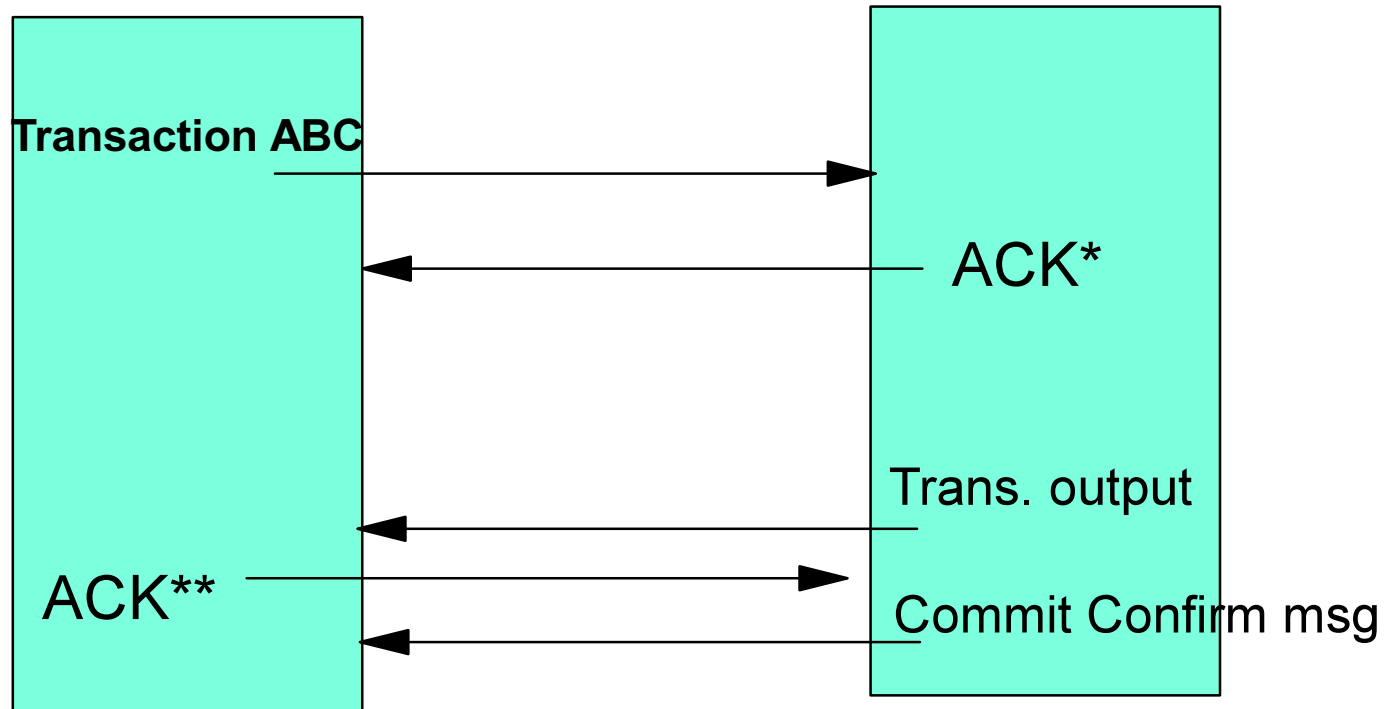


MQSeries uses Client-Bid with Resync. The rest of the OTMA clients used Client-Bid.

# Basic Message Protocol

OTMA Client

IMS



ACK\* will be sent if the OTMA client requests.

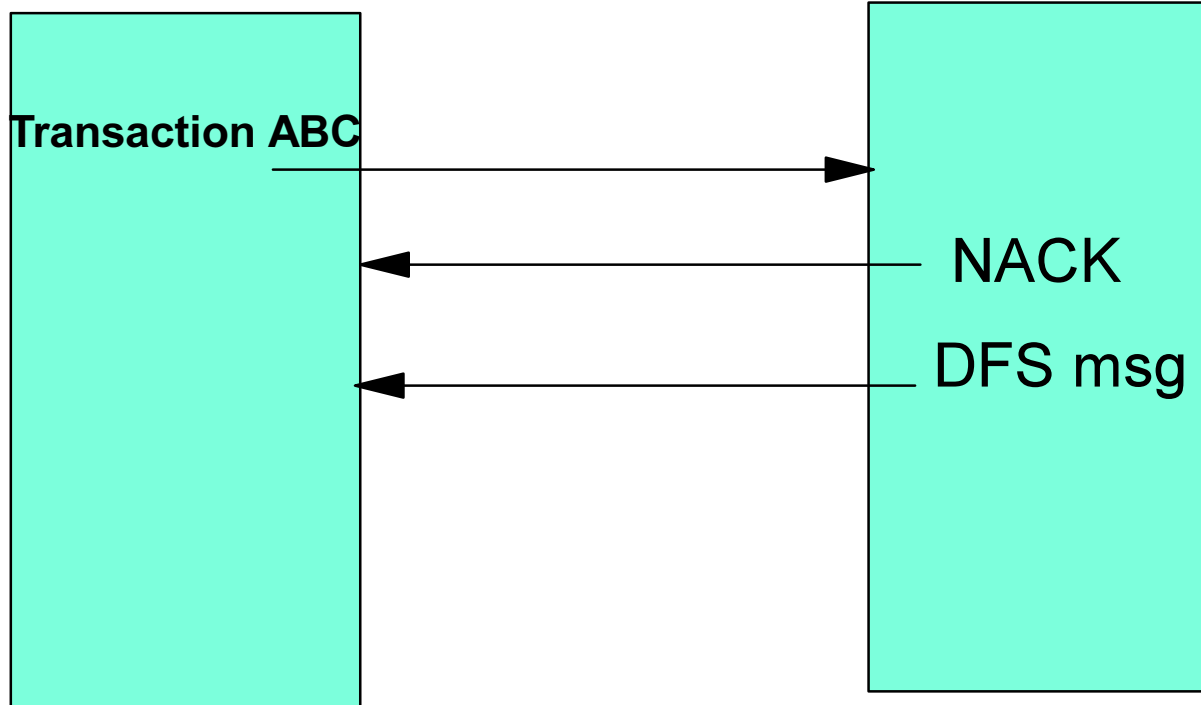
ACK\*\* will not be sent when send-then-commit with Syncleve = NONE.

Commit confirm msg exists only for send-then-commit messages.

# Basic Message Protocol...

OTMA Client

IMS



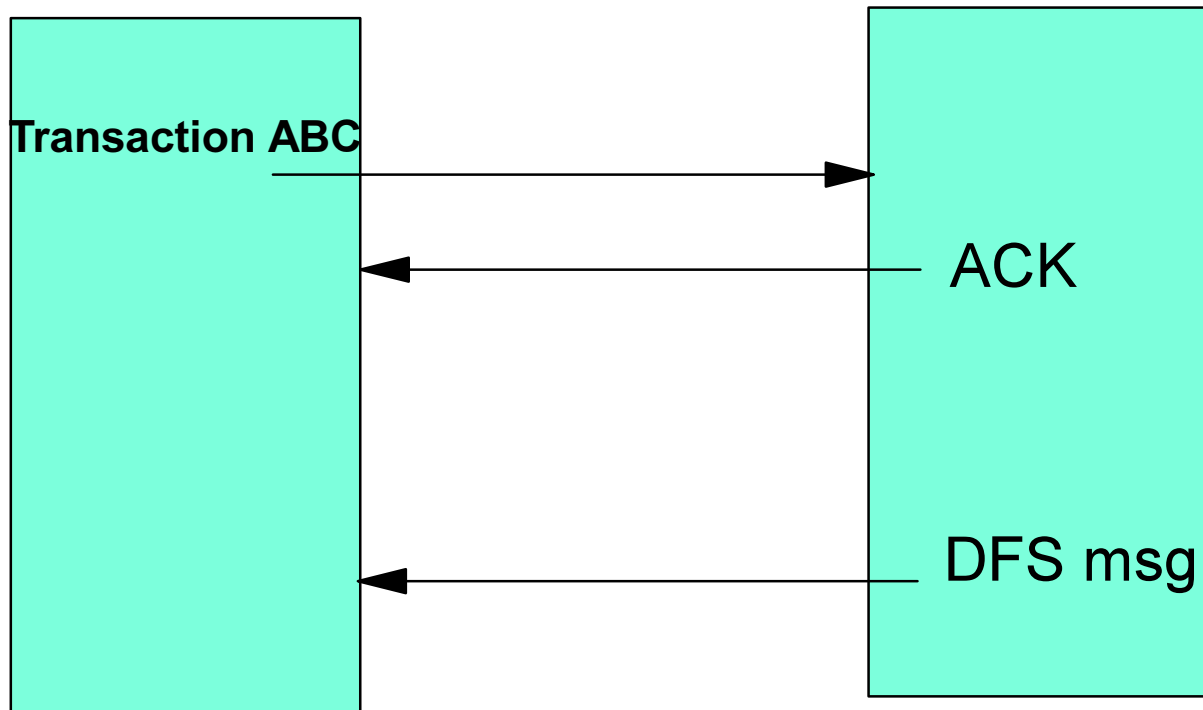
NACK will be sent if the message can not be inserted into IMS. If an IMS error message will also be issued to report the reason, the NACK message will say so in the prefix.



# Basic Message Protocol...

OTMA Client

IMS



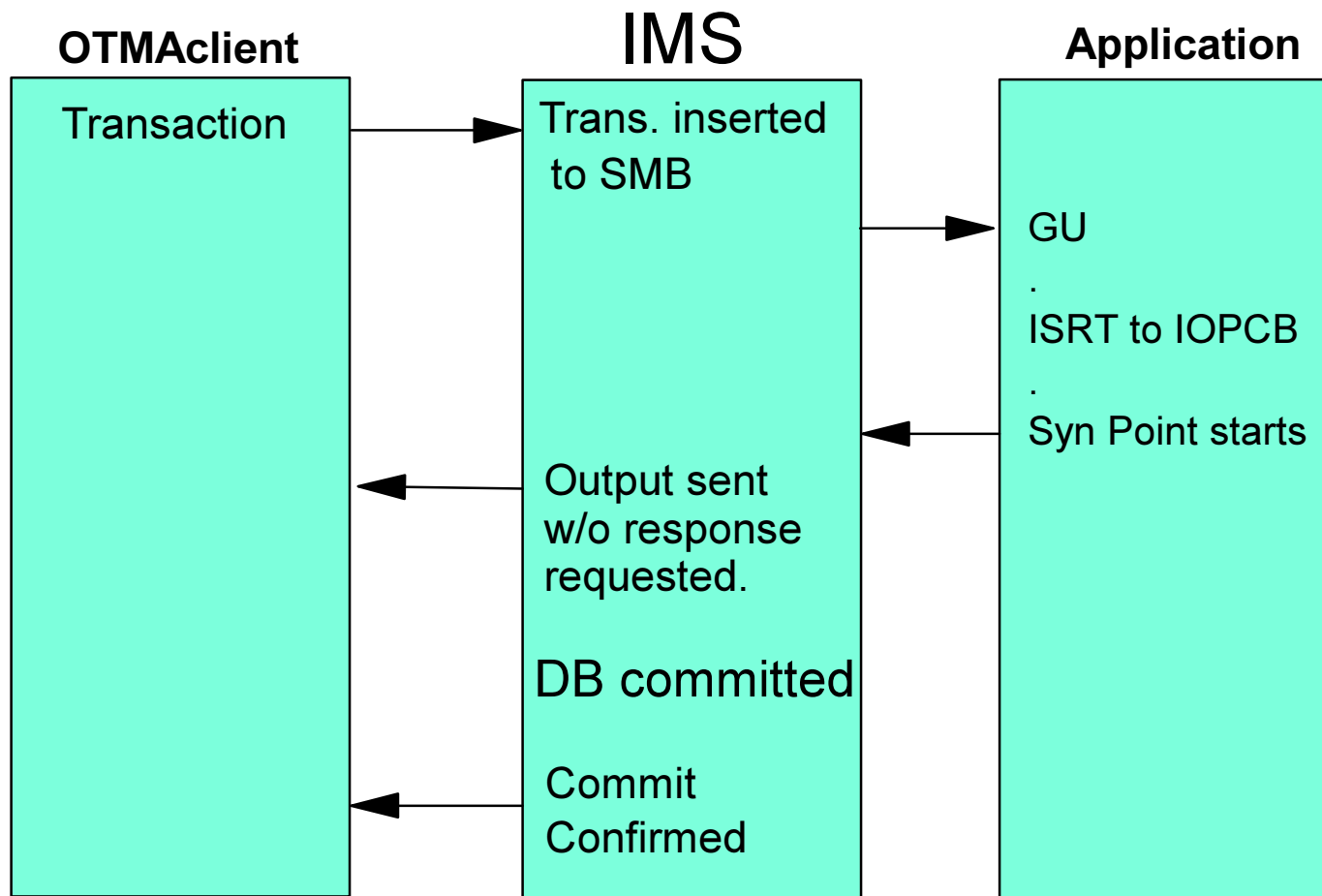
A DFS error message will be issued if the application abends.

# Commit Mode ???

- **Send-then-Commit (Commit mode 1)**
  - ▶ Synclevel = None
  - ▶ Synclevel = Confirm
  - ▶ Synclevel = SyncPt
- **Commit-then-Send (Commit Mode 0)**
  - ▶ Synclevel = Confirm

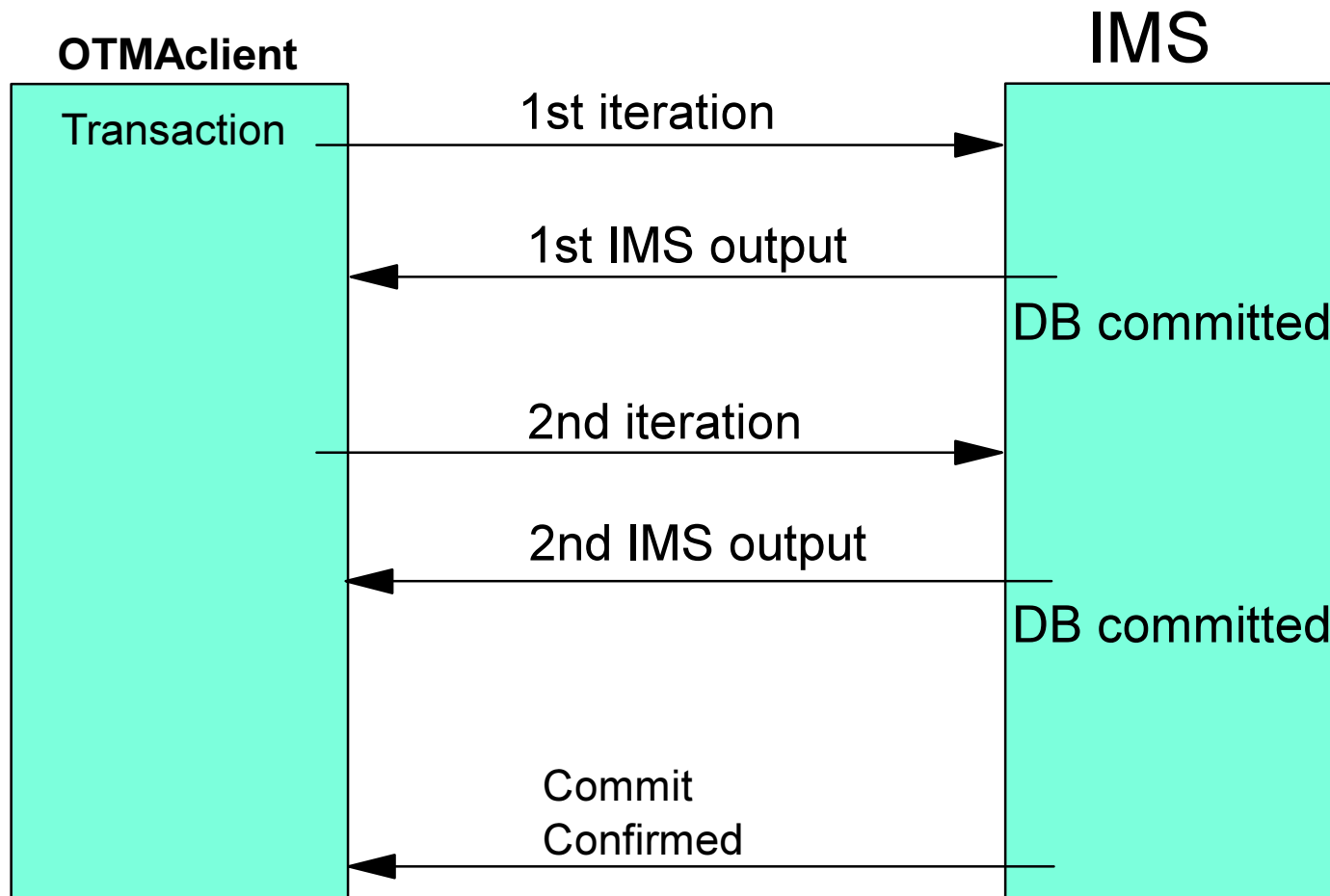
# Send-then-commit message

(synch level = NONE)



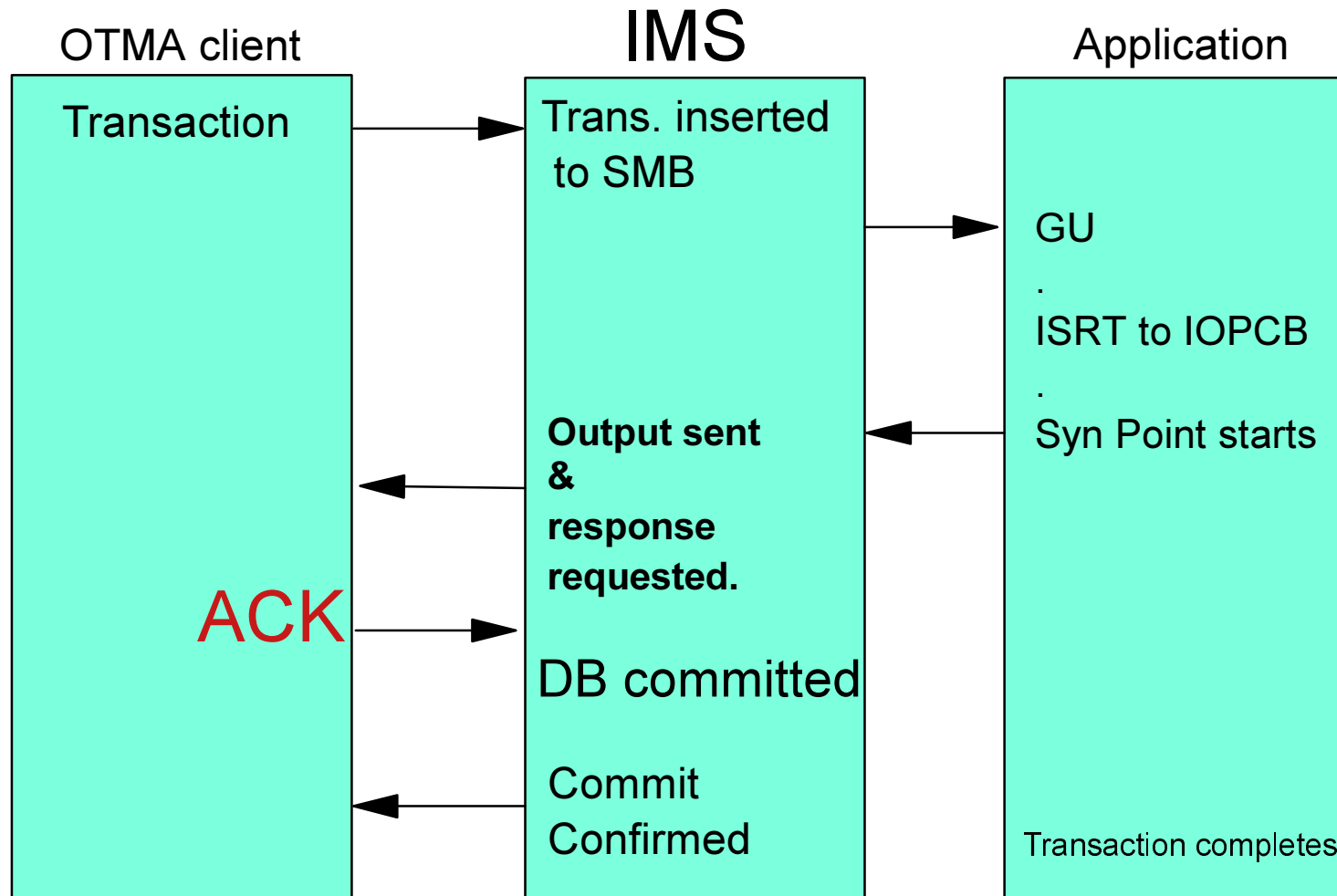
# Send-then-commit message

(synch level = NONE, for IMS Conversational trans)



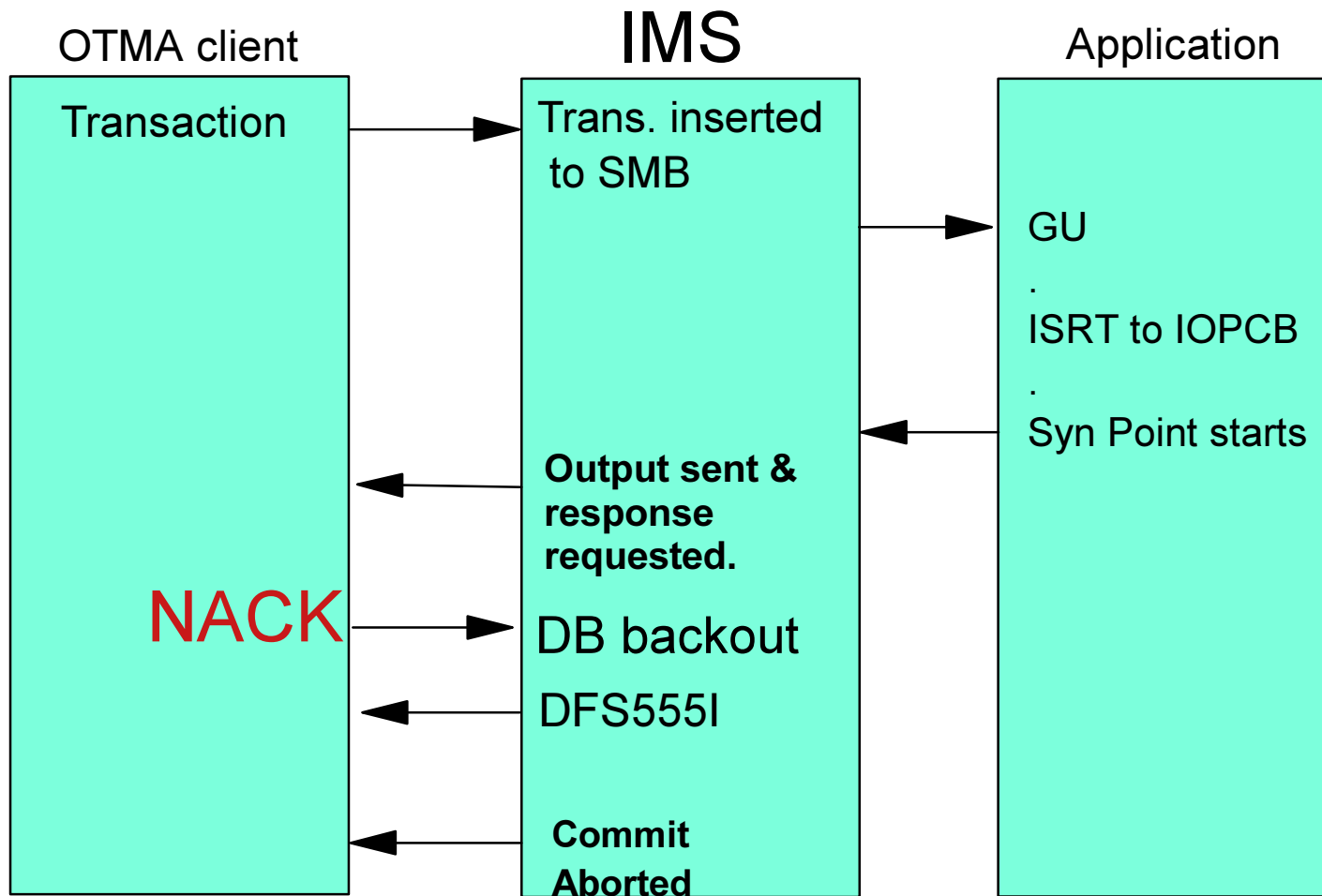
# Send-then-commit message

(synch level = CONFIRM)



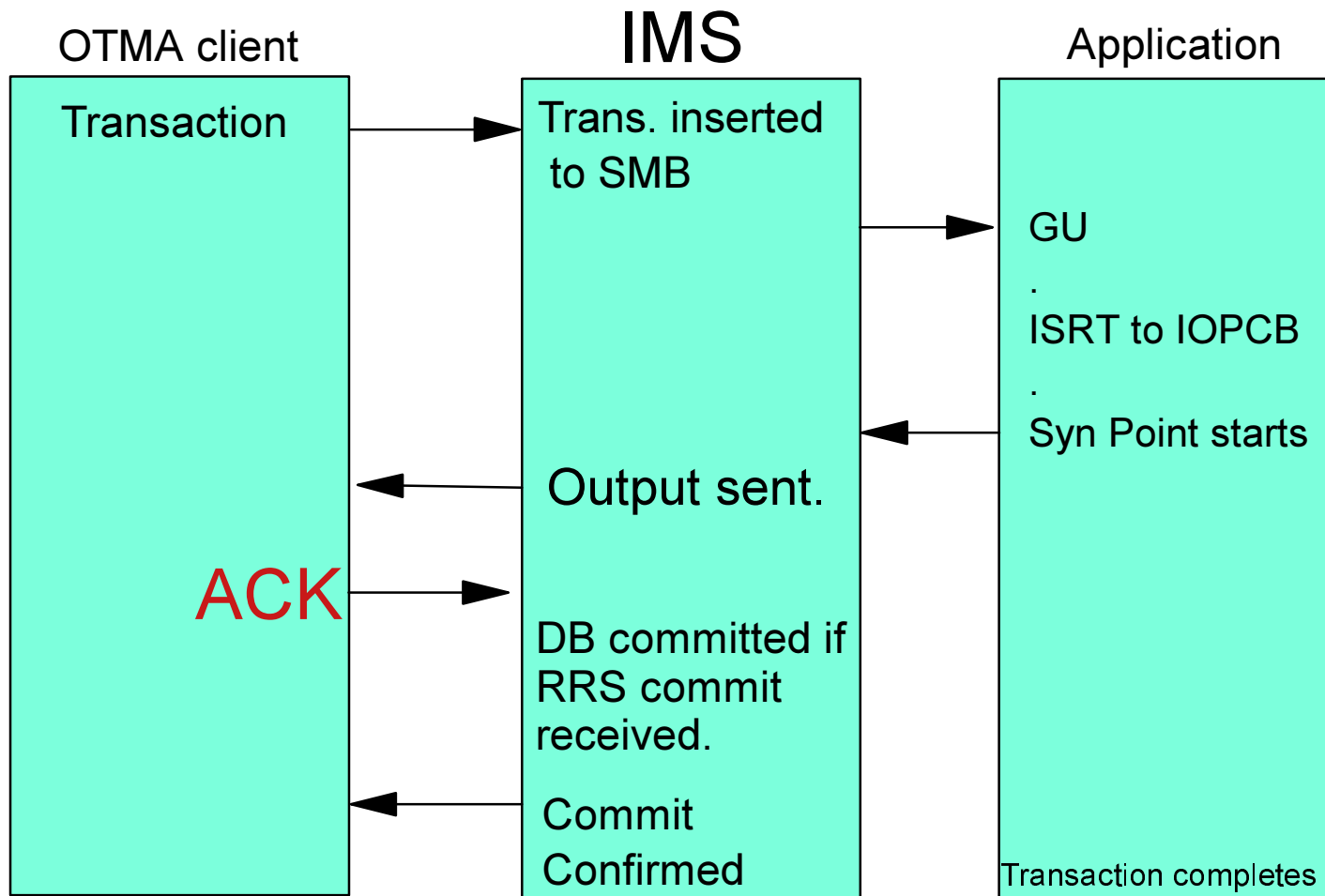
# Send-then-commit message

(synch level = CONFIRM)

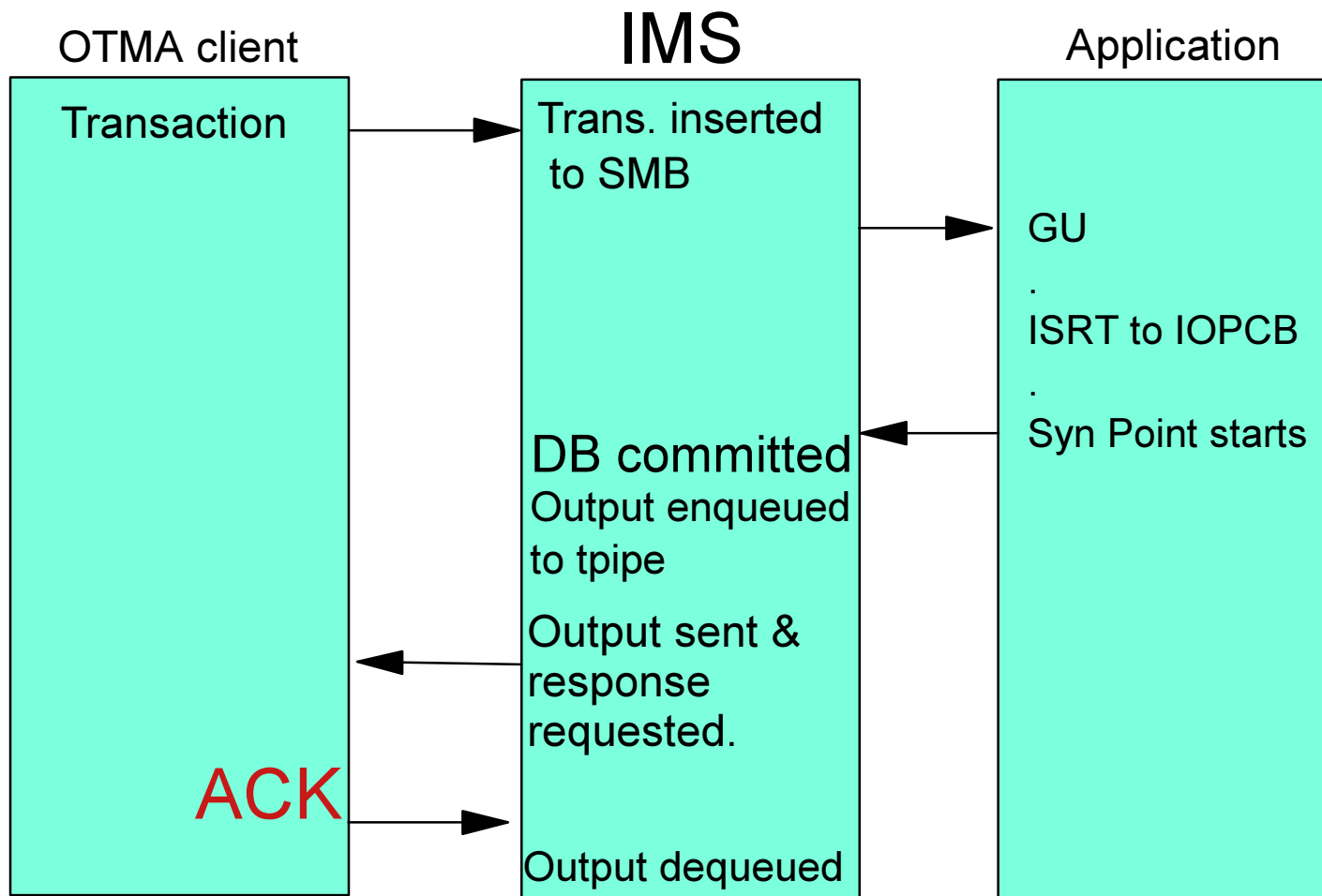


# Send-then-commit message

(synch level = SYNCPT)

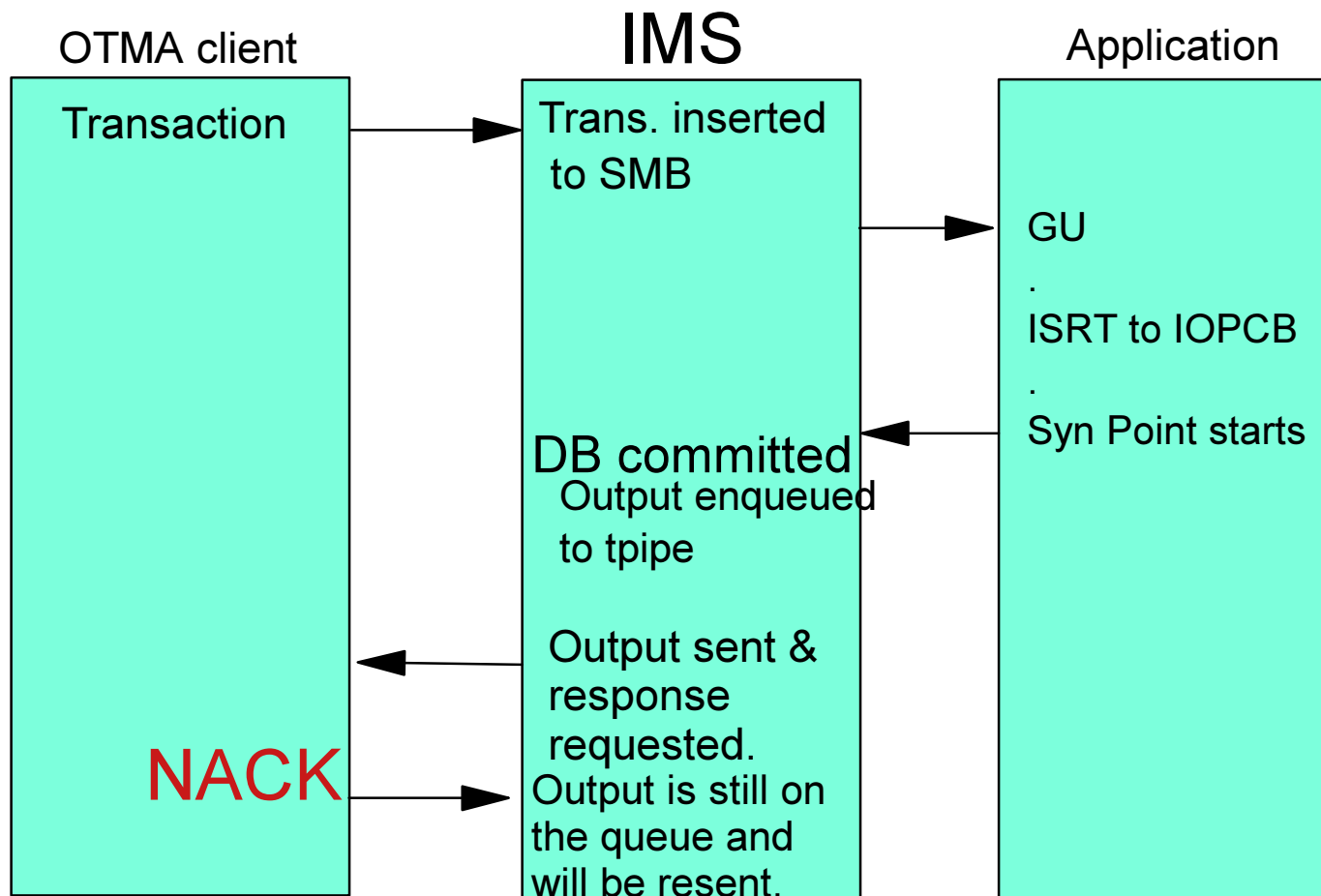


# Commit-then-send message





# Commit-then-send message



# Program Switches

- If IMS detects that the input OTMA message is Commit-then-send(CM0), then the IMS output message will always be the Commit-then-send (CM0).
- If the input OTMA message is Send-then-commit (CM1), then the IMS output message will in general be the Send-then-commit (CM1) except the following cases involved program switches.

# Program Switches...

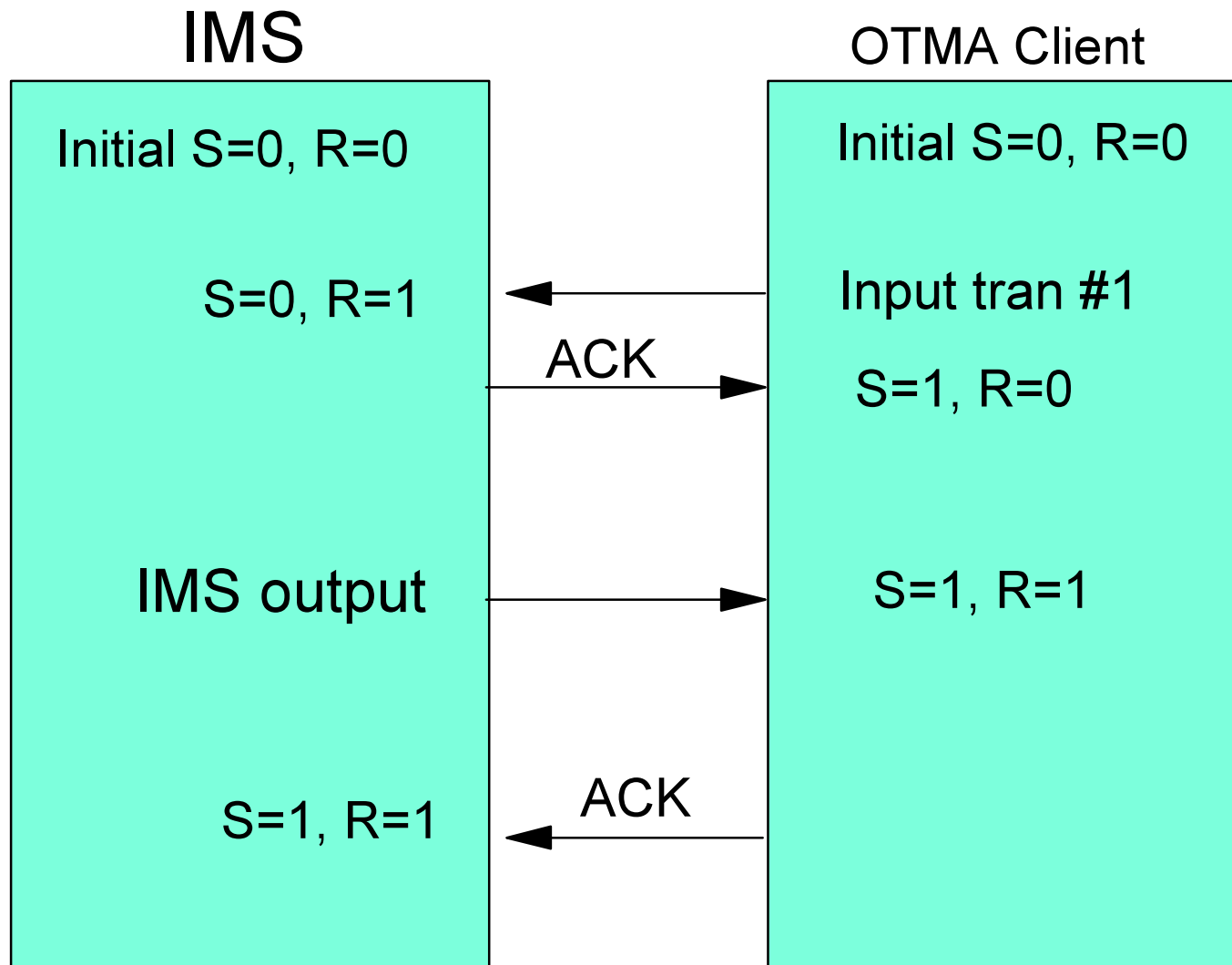
- If an OTMA send-then-commit (CM1) transaction does not send a response back to the client, then performs a program-to-program switch using an express PCB, the response from the second transaction will be processed using the Commit-then-send (CM0).

# Program Switches...

- Also, if an OTMA send-then-commit (CM1) transaction does not send a response back to the client, then performs multiple program-to-program switches using non-express PCB's, the response from one or more of the subsequent transactions may be processed using commit-then-send (CM0).

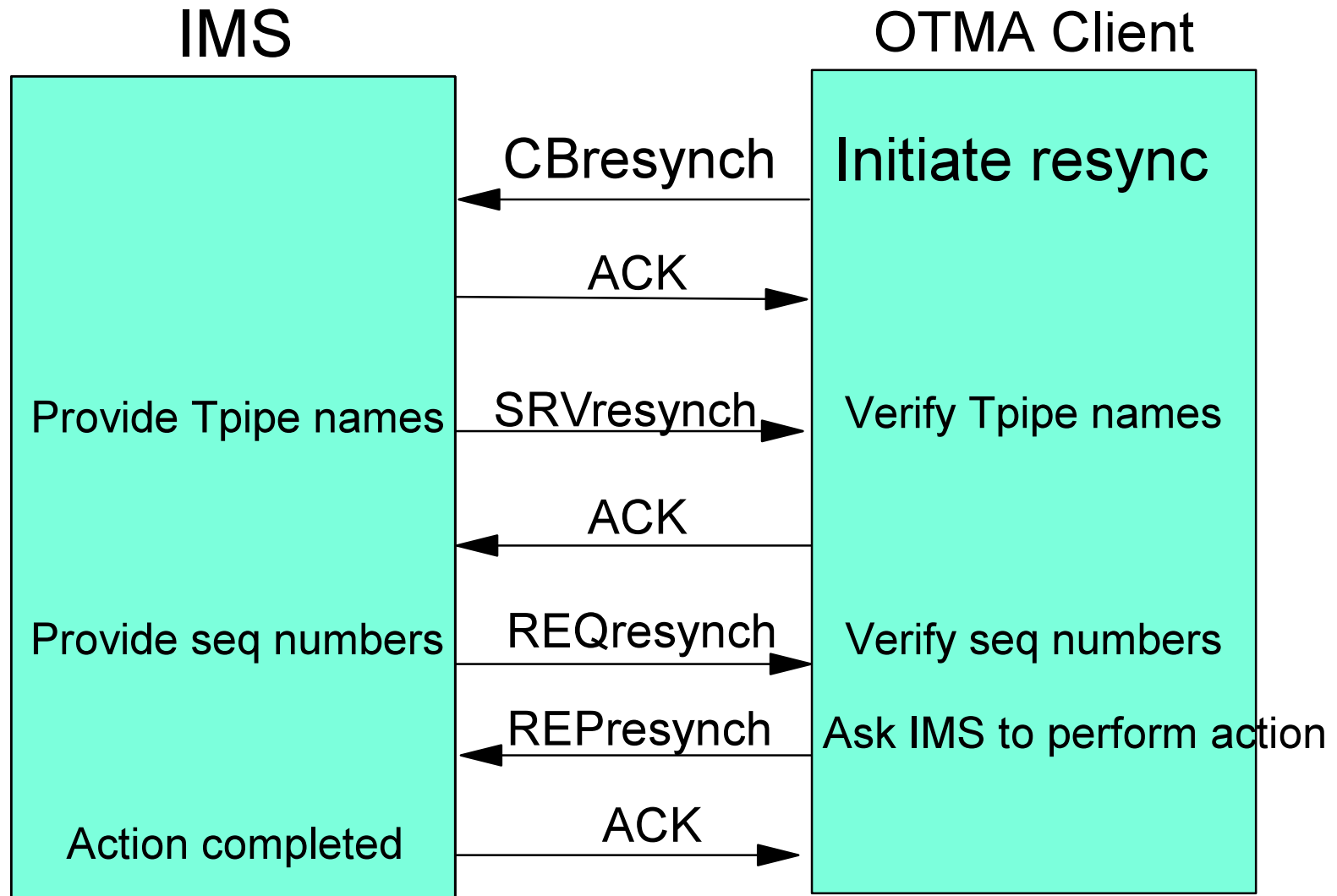
# Resynch protocol

For every input/output message



# Resynch protocol

During connection time



R=1

# Resync Protocol

- Action requested by REPresync

| Action        | Meaning  |
|---------------|--|
| Continue      | Seg # are correct in both sides  |
| Dequeue       | Client's receive count is 1 high than the IMS send count.              |
| Reset         | Client ask for sequence number reset                                   |
| Stop Tpipe    | Client ask for stopping the tpipe                                      |
| Stop and Wait | Client's send and receive counts are so different from the IMS counts. |

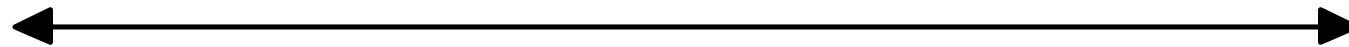
# Resync Protocol

- Action requested by REPresync
  - ▶ Continue: everything is good. Seq# matches.
  - ▶ **Dequeue last output**: ask IMS to dequeue the last output message. IMS may fail to perform the operation and leave the tpipe in **DQF** state. Need IMS dump and log records to trace the problem.
  - ▶ Reset Seq #: reset the send and receive seq numbers.
  - ▶ Stop Tpipe: make Tpipe unusable.
  - ▶ **Stop and Wait for TBresync**: the client detects that the sequence numbers are so different between the client and IMS. The IMS Tpipe in **TBR** state is unusable. Need IMS and MQSeries dump and log records to trace the problem. However, issuing MQSeries RESET command can reset the sequence numbers to make tpipe usable.



# OTMA Message Prefix

Prefix



Tpipe name  
Message type  
Sequence #  
Response indicator  
Chaining indicator

Commit mode  
Sync Level  
(different formats for protocol cmd)

Userid  
Group name  
Utoken

OTMA  
client data

# OTMA Prefix: Message Control Information

| Byte in Hex | Length | Content        | Value | Meaning  |
|-------------|--------|----------------|-------|--|
| 0           | 1      | Arch. Level    | X'01' | OTMA level   |
| 1           | 1      | Msg type       |       |  |
|             |        | Data           | X'80' | IMS output data or IMS Conv. Tran input iteration                |
|             |        | Trans Input    | X'40' | IMS trans or IMS command   |
|             |        | Response       | X'20' | ACK or NACK  |
|             |        | Command        | X'10' | OTMA protocol command  |
|             |        | Commit Msg     | X'08' | Commit-confirmation msg or a client's "IMS conv. cancel" request |
| 2           | 1      | Resp flag      |       |  |
|             |        | ACK            | X'80' | ACK  |
|             |        | NACK           | X'40' | NACK with sense code   |
|             |        | Resp Requested | X'20' | Either the client or server ask for an acknowledge (NACK or ACK) |
| 3           | 1      | Commit flag    |       |  |
|             |        | Committed      | X'80' | DB Committed for CM1   |
|             |        | Aborted        | X'40' | DB aborted for CM1   |

# OTMA Prefix: Message Control Information...

| Byte in Hex | Length | Content         | Value | Meaning                         |
|-------------|--------|-----------------|-------|---------------------------------|
| 4           | 1      | Command type    |       | OTMA protocol command type      |
| 5           | 1      | Process flag    |       |                                 |
|             |        | Msg in que      | X'08' |                                 |
|             |        | Sync Tpipe      | X'40' | MQseries turns on this flag     |
|             |        | Asyc output     | X'20' | unsolicited msg sent by IMS     |
|             |        | Error msg       | X'10' | An error msg followed a NACK    |
| 6           | 8      | Tpipe name      |       |                                 |
| E           | 1      | Chain flag      |       | multi-seg msg or single-seg msg |
|             |        | First in chain  | X'80' | first segment                   |
|             |        | Middle in chain | X'40' | IMS OTMA did not check it.      |
|             |        | Last in chain   | X'20' | last segment                    |

# OTMA Prefix: Message Control Information...

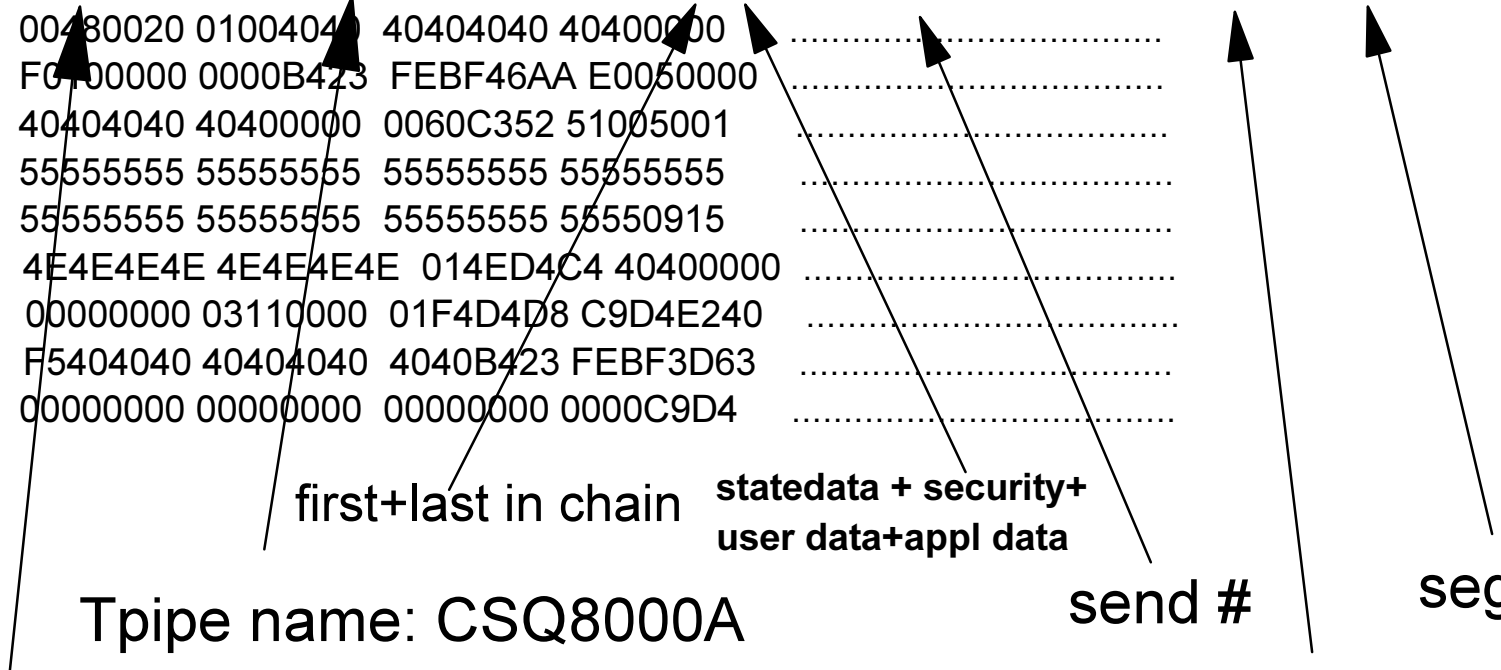
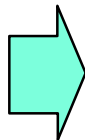
| Byte in Hex | Length | Content                | Value | Meaning                                    |
|-------------|--------|------------------------|-------|--|
| F           | 1      | Prefix flag            |       |  |
|             |        | State Data             | X'80' | State data in the prefix                   |
|             |        | Security               | X'40' | Security data in the prefix                |
|             |        | User Data              | X'20' | Client user data in the prefix             |
|             |        | Appl. Data             | X'10' | Application data in the message            |
| 10          | 4      | Send Seq #             |       | Send sequence number. Not used for resynch |
| 14          | 2      | Sense Code             |       | Accompanies a NACK message                 |
| 16          | 2      | Reason Code            |       | Accompanies a NACK message                 |
| 18          | 4      | recoverable sequence # |       | Send sequence number for resynch.          |
| 1C          | 2      | Seg #                  |       | segment # for multi-seg message            |
| 1E          | 2      | Reserved               |       |  |

# Message Control Info in 01 Record

## 01 RECORD

```

00000000 03F30000 01D18194 08000002 08000002 .....
00000020 4F921E05 E2E8E2F3 40404040 B423FEBF .....
00000040 00408100 C8000000 00000000 00000000 .....
00000060 FDFFFFFFFF 05860C60 C1D7D6D3 F1F14040 .....
00000080 00108600 0354FC00 00000000 00000000 .....
000000A0 F0F0F0C1 C1D7D6D3 F1F14040 B423FEBF .....
000000C0 40404040 40404040 40404040 00000000 .....
000000E0 01400000 0000C3E2 D8F8F0F0 F0C1A0F0 00000003 00000000 00000000 00010000
00000100 00480020 01004040 40404040 40400000 .....
00000120 F0700000 0000B423 FEBF46AA E0050000 .....
00000140 40404040 40400000 0060C352 51005001 .....
00000160 55555555 55555555 55555555 55555555 .....
00000180 55555555 55555555 55555555 55550915 .....
000001A0 4E4E4E4E 4E4E4E4E 014ED4C4 40400000 .....
000001C0 00000000 03110000 01F4D4D8 C9D4E240 .....
000001E0 F5404040 40404040 4040B423 FEBF3D63 .....
00000200 00000000 00000000 00000000 0000C9D4 .....
  
```



# OTMA Prefix:

## State Data for transaction input

| Byte in Hex | Length | Content              | Value | Meaning  |
|-------------|--------|----------------------|-------|--|
| 0           | 2      | Length               |       | Length of the state data   |
| 2           | 1      | Server State         |       |  |
|             |        | Conversational State | X'80' | Set by IMS to indicate that the IMS conversation is active. The client needs to keep this flag for the subsequent message interaction. Only IMS can reset this flag. |
|             |        | Special Queue        | X'20' | The message was sent from a special message pipe queue.  |
| 3           | 1      | Sync flag            |       | Commit mode 1 or 0   |
|             |        | Commit-then-send     | X'40' | CM0  |
|             |        | Send-then-commit     | X'20' | CM1  |
| 4           | 1      | SyncLevel            |       |  |
|             |        | None                 | X'00' |  |
|             |        | Confirm              | X'01  |  |
|             |        | Syncpt               | X'02' |  |

# OTMA Prefix:

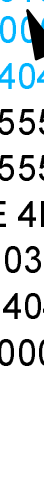
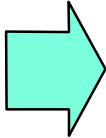
## State Data for transaction input...

| Byte in Hex | Length | Content        | Value | Meaning   |
|-------------|--------|----------------|-------|---|
| 5           | 1      | Reserved       |       |   |
| 6           | 8      | Map name       |       | MOD name  |
| E           | 16     | Server Token   |       | Server token for IMS control block. Must be returned by the client. |
| 1E          | 16     | Cor. Token     |       | Any token prepared by the client. IMS will not change it.           |
| 2E          | 16     | Context ID     |       | RRS Context ID for Synclevel = SYNCPT                               |
| 3E          | 8      | LTERM override |       | Lterm override  |
| 46          | 2      | Reserved       |       |   |
|             |        |                |       |   |

# State Data in 01 Record

## 01 RECORD

```
00000000 03F30000 01D18194 08000002 08000002 .....
00000020 4F921E05 E2E8E2F3 40404040 B423FEBF .....
00000040 00408100 C8000000 00000000 00000000 .....
00000060 FDFFFFFF 05860C60 C1D7D6D3 F1F14040 .....
00000080 00108600 0354FC00 00000000 00000000 .....
000000A0 F0F0F0C1 C1D7D6D3 F1F14040 B423FEBF .....
000000C0 40404040 40404040 40404040 00000000 .....
000000E0 01400000 0000C3E2 D8F8F0F0 F0C1A0F0 .....
00000100 00480020 01004040 40404040 40400000 00000000 00000000 00000000 00007E41
00000120 F0100000 0000B423 FEBF46AA E0050000 00000000 00000000 00000000 00004040
00000140 40404040 40400000 0060C352 51005001 .....
00000160 55555555 55555555 55555555 55555555 .....
00000180 55555555 55555555 55555555 55550915 .....
000001A0 4E4E4E4E 4E4E4E4E 014ED4C4 40400000 .....
000001C0 00000000 03110000 01F4D4D8 C9D4E240 .....
000001E0 F5404040 40404040 4040B423 FEBF3D63 .....
00000200 00000000 00000000 00000000 0000C9D4 .....
```

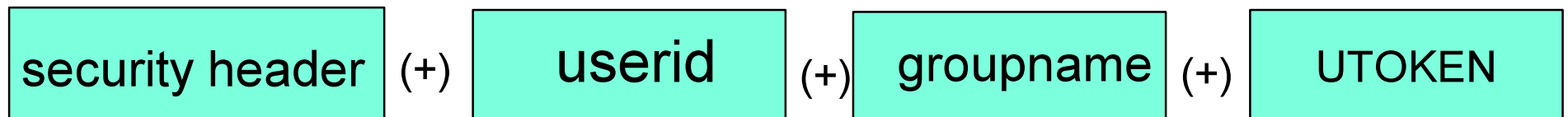


Send-then-commit

Sync Level = Confirm



# OTMA Prefix: Security Data header



| Byte in Hex | Length | Content       | Value | Meaning                       |
|-------------|--------|---------------|-------|-------------------------------|
| 0           | 2      | Length        |       |                               |
| 2           | 1      | Security flag |       | For /SEC OTMA PROFILE only    |
|             |        | No Security   | N     | No RACF security checking     |
|             |        | Check         | C     | Perform RACF checking in CTL  |
|             |        | Full          | F     | Check + MPR security checking |
| 3           | 1      | Reserved      |       |                               |

# OTMA Prefix: Security (userid, group name, UTOKEN)

| Byte in Hex | Length | Content                | Value | Meaning                          |
|-------------|--------|------------------------|-------|----------------------------------|
|             | 1      | Userid Length          |       | Length of the following 2 fields |
|             | 1      | Type                   | X'02' | Identify this is for Userid      |
|             | 1-8    | userid                 |       | 1 to 8 bytes of userid           |
|             | 1      | RACF group name length |       | Length of the following 2 fields |
|             | 1      | Type                   | X'03' | Identify this is for RACF group  |
|             | 1-8    | group name             |       | 1 to 8 bytes of group name       |
|             | 1      | Utoken length          |       | Length of the following 2 fields |
|             | 1      | Type                   | X'00' | Identify this is for Utoken      |
|             | 1-80   | RACF user token        |       | 1-80 bytes of UTOKEN             |

# Security Data in 01 Record

## 01 RECORD

```
00000000 03F30000 01D18194 08000002 08000002 .....
00000020 4F921E05 E2E8E2F3 40404040 B423FEBF .....
00000040 00408100 C8000000 00000000 00000000 .....
00000060 FFFFFFFF 05860C60 C1D7D6D3 F1F14040 .....
00000080 00108600 0354FC00 00000000 00000000 .....
000000A0 F0F0F0C1 C1D7D6D3 F1F14040 B423FEBF .....
000000C0 40404040 40404040 40404040 00000000 .....
000000E0 01400000 0000C3E2 D8F8F0F0 F0C1A0F0 .....
00000100 00480020 01004040 40404040 40400000 .....
00000120 F0100000 0000B423 FEBF46AA E0050000 .....
00000140 40404040 40400000 0060C352 51005001 80551555 95555555 55555555 55555555
00000160 55555555 55555555 55555555 55555555 80551555 95555555 55555555 55555555
00000180 55555555 55555555 55555555 55550915 55555555 55555555 55555555 55555555
000001A0 4E4E4E4E 4E4E4E4E 014ED4C4 40400000 .....
000001C0 00000000 03110000 01F4D4D8 C9D4E240 .....
000001E0 F5404040 40404040 4040B423 FEBF3D63 .....
00000200 00000000 00000000 00000000 0000C9D4 .....
```

Security prefix

The total sec data

length is x'60'

type=utoken

length of following 2 fields

UTOKEN

# OTMA Prefix:

## OTMA user data (Client's data, not for IMS application data)

| Byte in Hex | Length | Content       | Value    | Meaning                                     |
|-------------|--------|---------------|----------|---|
| 0           | 2      | Length        |          | Length of client's user data + length field |
| 2           | 1-1022 | client's data | anything | The OTMA user data section is optional.     |

# IMS 6701 Record for OTMA

Input. Could be TIB1

INTERNAL TRACE RECORD

ID = TIB0 SEGNO=00 RECNO = 000000E5

YTIB

065DCEB0 000000 E8E3C9C2 E8E3C9C2 00000000 065DCC60 0665E120 E3D7

TPIPE

0665E120 000000 C3D3C9C5 D5E3F140 40404040 40404040 E3D7C9D7 C5F14040 F99A1F90 00000000

0665E140 000020 F99A1F28 00000000 00000000 05C24908 065DCEB0 066602B0 00000001 00000000

0665E160 000040 80000000 00000000 00000002 00000000 00000001 00000000 00000000 0665E070

0665E180 000060 0665E0D8 00000000 00000000 00000000 00000000 00000000 00000000 00000000

0665E1A0 000080 00000000 00000000

MSG PREF

065CFD08 000000 010E0000 0040D581 84D52000 E3D7C9D7 C5F14040 C1D7D6

065CFD28 000020 00698B07 00000000 065DCC60 C3D3C9C5 D5E3F140 404040

065CFD48 000040 00000000 00000000 00000000 00000000 01400000 0000E3D7 C9D7C5F1 4040A0D0

065CFD88 000080 00000000 00000000 00000000 0000C9D4 00480020 0000E3C5 E2E3D4C1 D7400000

065CFDA8 0000A0 40404040 40404040 40404040 40404040 E2F0F0F0 F0F10000 00000000 00004040

065CFDC8 0000C0 80551555 95555555 55555555 55555555

065CFDE8 0000E0 55555555 55555555 55555555 55555555

065CFE08 000100 15151515 15151515 15151515 15150000

I/O BUFF

06668AF0 000000 00380000 C1D7D6D3 F1F14040 D6E3D4C1 40A39640 D4D7D7

06668B10 000020 40404040 40404040 40404040 40404040 40404040 404040

IMS appl. data

rec count

send count

tpipe name

OTMA prefix, starting x'50'



# IMS 6701 Record for OTMA

CM1 output, could be SLM1

```

INTERNAL TRACE RECORD      ID = SLM0  SEGNO=00  RECNO = 000000E5
YTIB
065DCEB0 000000  E8E3C9C2 E8E3C9C2 00000000 065DCC60 0665E120 E3D7
TPIPE
0665E120 000000  C3D3C9C5 D5E3F140 40404040 40404040  E3D7C9D7 C5F14040  F99A1F90 00000000
0665E140 000020  F99A1F28 00000000 00000000 05C24908 065DCEB0 066602B0 00000001 00000000
0665E160 000040  80000000 00000000 00000002 00000000 00000001 00000000 00000000 0665E070
0665E180 000060  0665E0D8 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0665E1A0 000080  00000000 00000000
MSG PREF
065CFD08 000000  010E0000 0040D581 84D52000 E3D7C9D7 C5F14040 C1D7D6
065CFD28 000020  00698B07 00000000 065DCC60 C3D3C9C5 D5E3F140 404040
065CFD48 000040  00000000 00000000 00000000 00000000 01800000 0000E3D7 C9D7C5F1 4040A0D0
065CFD88 000080  00000000 00000000 00000000 0000C9D4 00480020 0000E3C5 E2E3D4C1 D7400000
065CFDA8 0000A0  40404040 40404040 40404040 40404040 E2F0F0F0 F0F10000 00000000 00004040
065CFDC8 0000C0  80551555 95555555 55555555 55555555
065CFDE8 0000E0  55555555 55555555 55555555 55555555
065CFE08 000100  15151515 15151515 15151515 15150000
I/O BUFF
06668AF0 000000  00380000 C1D7D6D3 F1F14040 D6E3D4C1 40A39640 D4D7D7
06668B10 000020  40404040 40404040 40404040 40404040 40404040 404040
  
```

IMS appl. data

OTMA prefix for output,  
starting x'50'

rec count  
for CM0

send count  
for CM0

tpipe name



# IMS 6701 Record for OTMA

CM0 output, could be QAB1

```

INTERNAL TRACE RECORD      ID = QAB0  SEGNO=00  RECNO = 000000E5
YTIB
065DCEB0 000000  E8E3C9C2 E8E3C9C2 00000000 065DCC60 0665E120 E3D7
TPIPE
0665E120 000000  C3D3C9C5 D5E3F140 40404040 40404040  E3D7C9D7 C5F14040  F99A1F90 00000000
0665E140 000020  F99A1F28 00000000 00000000 05C24908 065DCEB0 066602B0 00000001 00000000
0665E160 000040  80000000 00000000 00000002 00000000 00000002 00000000 00000000 0665E070
0665E180 000060  0665E0D8 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0665E1A0 000080  00000000 00000000
MSG PREF
065CFD08 000000  010E0000 0040D581 84D52000 E3D7C9D7 C5F14040 C1D7D6
065CFD28 000020  00698B07 00000000 065DCC60 C3D3C9C5 D5E3F140 404040
065CFD48 000040  00000000 00000000 00000000 00000000 01800000 0040E3D7 C9D7C5F1 4040A0D0
065CFD88 000080  00000000 00000000 00000003 0000C9D4 00480040 0100E3C5 E2E3D4C1 D7400000
065CFDA8 0000A0  40404040 40404040 40404040 40404040  E2F0F0F0 F0F10000 00000000 00004040
065CFDC8 0000C0  80551555 95555555 55555555 55555555
065CFDE8 0000E0  55555555 55555555 55555555 55555555
065CFE08 000100  15151515 15151515 15151515 15150000
I/O BUFF
06668AF0 000000  00380000 C1D7D6D3 F1F14040 D6E3D4C1 40A39640 D4D7D7
06668B10 000020  40404040 40404040 40404040 40404040 40404040 404040
  
```

IMS appl. data

OTMA prefix for CM0 output, starting x'50'

rec count

New send count

send count

tpipe name

Sync tpipe

IMS Technical Conference



# IMS 6701 Record for OTMA

- To activate the record:
  - ▶ Issue /TRA SET ON Tmember xxx Tpipe yyy
  - ▶ For an input message, the record is cut before the message is inserted into the queue. Also it's cut for the first input segment processed by DFSYIOE0 user exit.
  - ▶ For an output message, the record is cut before the message is sent to the OTMA client. Also it's cut for the first output segment processed by the DFSYIOE user exit.



# Summary

- Every OTMA message for IMS transaction consists of OTMA prefix and application data.
- OTMA prefix controls how IMS processes the message.
- There are OTMA protocol commands to provide unique handshaking between OTMA client and server. For example: Client-bid, REQresync, REPresync, Suspend tpipe, Resume Tpipe..
- IMS 6701 record for OTMA has OTMA tpipe info, prefix data, and input/output data.